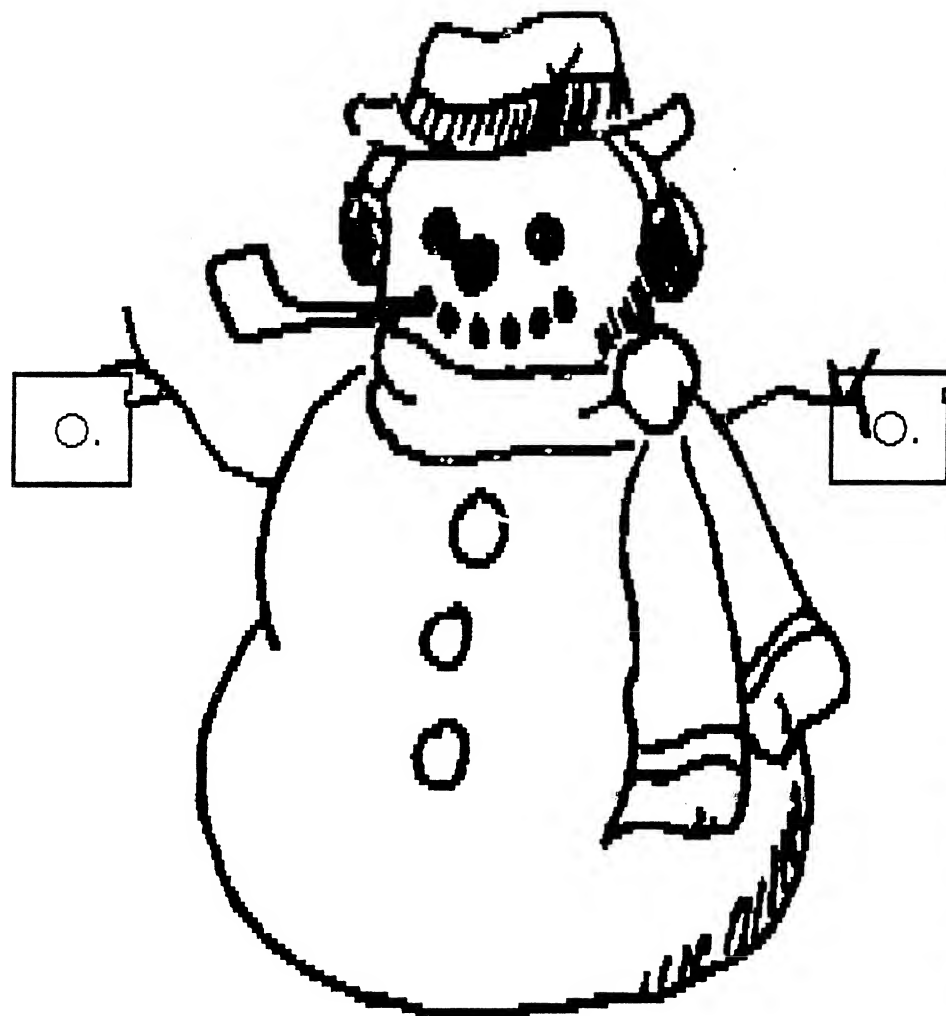


C. H. G. S.

M
O
N
I
T
O
R



"Editor preparing this Vol.3, Issue #1"

January '88

OBLIGATORY STUFF

PRESIDENT	Richard Maze
VICE-PRESIDENT	Barry Bircher
TREASURER	Harry Chong
LIBRARIAN	Earl Brown
ASS'T LIBRARIAN	Steve Bogues
EDITOR	Ken Danylcuk
ASS'T EDITOR	Greg Rezansoff
MEMBER AT LARGE	Gord Williams

THE MONITOR is published monthly by the COMMODORE USERS' GROUP OF SASKATCHEWAN (CUGS), Regina, Sask., Canada. CUGS meetings are held at 7 pm on the first Wednesday of every month (unless otherwise noted) in the North-West Leisure Centre, corner of Rochdale Boulevard and Arnason Street.

Anyone interested in computing, especially on the C64, 128 or 64C, is welcome to attend any meeting. Out of town members are also welcome, but may be charged a small mailing fee for newsletters. Members are encouraged to submit public domain software for inclusion in the CUGS DISK LIBRARY. These programs are made available to members. Any member is entitled to purchase DISKS from our public domain library for a nominal fee. Programs are 'freeware', from computer magazines, or the public domain. Individual members are responsible for deleting any program that he/she is not entitled to by law (you must be the owner of the magazine in which a particular program was printed). To the best of our knowledge, all such programs are identified in their basic listings. Please inform us should you find otherwise.

CUGS is a non-profit organization comprised of C64, 64C, C128, and 128D users interested in sharing ideas, programs, knowledge, problems and solutions with each other. The more members participate, the better the variety of benefits. Membership dues are pro-rated, based on a January to December year.

IN THIS ISSUE

JANUARY "JEMS"	- Leader's litany!
MEETING PLACE	- Date, Time, Place, Agenda
EDITORIAL	- 3 cheers 4 3rd parties!
SIR RICHARD'S BASIC	- One more in the SEquence!
SCRATCH 'N' SAVE	- Earl disk dilemma
NONSENSE NONOK!	- Neat, sweet ... a treat!
DISK DELIGHTS	- News, reviews and clues for youz!

Editorial:

Hooray for third parties!? So, what about the first party? The "third-party" I'm refering to are third-party SOFTWARE and HARDWARE DEVELOPERS. And Commodore owners probably more than other computer owners have reason to be grateful to these hi-tech entrepreneurs!

As I prepared this issue (and the up-coming meeting presentation), I was made keenly aware of the influence of third-party developers on computers in general and Commodore's in particular. "Third-party" is a reference to their position in the "chain of creativity" with respect to a computer. "First party" refers to the original developer of a particular machine (Commodore Business Machines, Tandy, Apple Corp., etc.). "Second-party" is the user. The manner in which a machine is designed dictates how quickly and keenly others not associated with the originating company or as

simple users begin to create software and hardware for a particular machine. Apple probably set the best example for "third-party" encouragement. The Apple was created as an "open book" - anyone who genuinely cared, could get details on the machine's innards the better to create useful soft- and hardware materials for it.

Commodore "sort of" followed their example, while Atari, Tandy and Texas Instruments preferred to keep their hi-tech tricks secret from all but their own private gurus. The results - Apple machines carved an incredible niche in the annals of home computing, with very little software produced by Apple itself! Others saw the potential of the machine and took it to limits even its "parent" doubted possible. Commodore followed suit (sort of)!

Apple creators ENCOURAGED others to create for their machine. Commodore sought people to create for them - they were a complete, vertically-integrated company, from plastic case to final product and wanted to keep it so! However, their leadership at the time (1980-82) sought to battle Apple, Atari and Texas Instruments for the home market, and concentrated on getting machines into homes. They did this by selling dreams - they showed prototype software as the norm and convinced millions that their machine could do marvellous things. The truth (as many a disappointed buyer will attest) was that the machine COULD do marvellous things IF THE RIGHT SOFTWARE WERE AROUND.

Commodore, (un)fortunately, had only moderate success in producing their own software. In an effort to give their new machines (VIC20, C64) credibility and strength in the home market, Commodore opened their secrets to the hi-tech wizards. The results are evident in any computer department or software store - thousands of software titles and hardware add-ons making the promise of the 64/128 (even the AMIGA) come true!!

So how did preparing articles on disk drives lead to this epistle on third-party developers? Commodore delivered on its first promise - an inexpensive, FRIENDLY (bet you haven't seen THAT word for a while!) powerful home computer - but at a price. Other computers, without exception, sacrificed some "friendliness" to the user to maximize speed and efficiency. Commodore decided to make their computer compensate for the natural carelessness of the under-educated user. No computer produced yet rivals Commodore's screen editor, or tape/disk reliability IN THE HANDS OF AN AMATEUR, or its incredible LACK of speed in data communication. Most of the "slowness" in Commodore machinery is due to deliberate software looping and multiple error-checking. (Hang in there, I've almost made my point!)

It was not 'til third-party developers began creating AROUND the deliberate slowness that the true power of these machines surfaced! I have a fast-format routine that formatted over 100 disks flawlessly at an average speed of 10.2 sec./disk! My high-power AT compatible takes 40 seconds. There are commercial copiers that will copy an entire disk (664 blocks, even with some errors) in less than 2 minutes ON A SINGLE DRIVE! These advances are the result of efforts by that last bastion of the original HACKER ethic - the third-party developer, whether commercial enterprise or dedicated amateur/hobbyist - anxious to push a machine to its absolute limit. More power to us (computer) people!

See ya next month!



January "Jems"

by Richard Maze

Welcome to the start of a new year for CUGS. I hope each of you had a good holiday season. Although we've not firmed up plans for this year, the executive have planned some agenda items for 1988. To start off, we will have a chance to look at the inner workings of a disk drive and see how to make some of the changes you have always wanted to make to your drive, and, more importantly, how to prevent some costly service calls. We'll also have a look at how files are stored on a diskette. The January meeting will also see the start of a new CUGS feature - the PRIZE DRAW. To be eligible to win all you need only be a PAID-UP MEMBER and be IN ATTENDANCE at the meeting. Although we haven't had a chance to finalize plans for other meetings, some ideas we are exploring are using a VCR with your C64, a software swap night, and the latest update of Earl's excellent INCOME TAX program.

The final outcome of our meeting agendas will be up to you. Tell an executive member what you'd like to see at meetings. If you have a program you'd like to present to the members, let us know. Also, what follow-up to presentations would you like to see?

The newsletter is an integral part of our club's operation. Please help out. If you don't want to give a presentation, at least WRITE A REVIEW OF A PROGRAM so other members can become familiar it. Also, use the newsletter to advertise hardware or programs you don't want any longer. Ken is doing a super job putting the newsletter together each month. The more articles he has, the easier his job will be. It is always easier to decide which article to use than to try and create something for a last half page.

Our disk library has been modified and rearranged. The task for this year is to increase the library. If you have public domain programs that are not in our library, donate them to the club and share them. Club policy is that we will trade club disks for your disks of new programs. The more extensive our library becomes the more everyone benefits. See our librarian, Earl, for any information about the disk library.

I am looking forward to 1988 as a banner year for CUGS. With the help of the executive and all of you, I would like to see us improve all aspects of our club. If everyone pitches in and helps whenever and wherever they can, we will have a super year and make our club stronger than ever.

UPCOMING CUGS MEETINGS FOR 1988

Wed. Feb. 3	Wed. May 4
Wed. Mar. 2	Wed. June 1
Wed. Apr. 6	

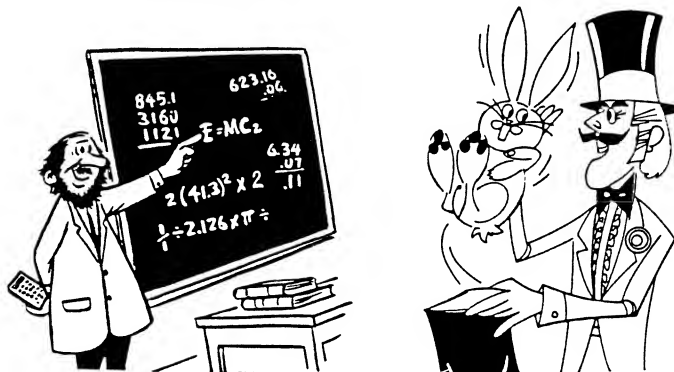
Write these on your calendar.

Like Grandma's Teeth On a Cold Winter Night!

Probably the most destructive copy protection devised commercially depended on Commodore's own built-in double-check overkill philosophy. If a disk error is encountered when the head is trying to read a sector, it tries to make sure it is really the disk at fault and sets about checking the position of the read/write head by telling the head to move "out" 45 tracks! The drive obediently tries, but gets stopped (abruptly, crudely and wrenchingly) by a flange on the head pulley meeting a physical stopper at the TRACK 0. Obliging, the ever-servile drive is told to RETRY this abominable action several times, presume the head is then on track 0 and try reading the (errored) track again! This whole scenario may repeat itself 2 or 3 times before the drive admits defeat!

This "head-knock" is the nemesis of 1540/1 drives. The violent physical abuse of the read/write head and other parts tended to make the disk drive (especially earlier versions) go out of alignment and be less than acceptably dependable. Unfortunately, a wealth of software was copy-protected by their originators with deliberately planted disk errors - forcing the head-knock. More sophisticated protection schemes and better drive construction have diminished the problem, but what about all that neat software that still plays a drum roll on your drive head? Type out and save a copy of this cute one-liner. Run it, and you'll safely (AND SILENTLY) load most coarse error-protected software. To my knowledge, this fix has been printed in several publications, all by different authors, and at least twice as a public domain offering.

```
10 OPEN15,8,15:PRINT#15,"M-W";CHRS(106)
   CHRS(0)CHRS(1)CHRS(133):CLOSE15
```



Meeting Place

AGENDA:

PRESIDENT'S COMMENTS - Richard Maze

ALL ABOUT DISKS (I) - How it works in a nutshell
(15 minutes or less!)

LOOKING AT A DISK! - Disk Doc Demo by Dick!

*****coffee****visiting****disk-picking*****

DISK DRIVE MAINTENANCE ON A small BUDGET

SEEING STARS *****

It doesn't take long for Commodore users to appreciate the speed and convenience of the 'wildcard' character (*) in loading programs off disk. Familiarity, unfortunately, does NOT mean understanding. The 'wildcard' character functions in several ways depending on it's use in context.

If your computer is newly-turned-on, the command `LOAD"*",8` loads the FIRST PROGRAM ON THE DISK into the computer.

If, however, you've had your computer on, and have loaded OTHER programs already the same command will RELOAD THE LAST PROGRAM LOADED. EVEN A 'SYS64738' RESET HAS NO EFFECT! Often, people who make this little mistake have CHANGED DISKS in the meantime, and the result is a frustrating flashing red error light on the drive. Examining the error, you would find a "FILE NOT FOUND" caused the blinking!

There is a dependable 'fix' which will let the "*" work the way you expect. The 'fix' is also suggested as a guarantee you won't be attacked by the "save and replace" bug. Simply use the command as follows:

```
LOAD":*",8
```

(or even better)

```
LOAD"O:*",8
```

The ':' or 'O:' force the drive to do a "self-initialization" (re-check its internal information about the disk it's using), and, in the process "cleans up" any information still resident from the last load.

If you really want to "clean up" the RAM in your drive to a newly-turned-on state, use `OPEN15,8,15,"U:"` or `OPEN15,8,15,"UJ"`, hit `<RETURN>` and type `CLOSE15` to properly close the file. This "cleanup" will even RESET a software altered drive number! It has NO effect on computer RAM.

Incidentally, unless you like frustration, DON'T EVER USE THE '*' FOR SAVING!

CUGS software review:

REVIEW: DIRECTORY ASSISTANCE

DIRECTORY ASSISTANCE is a utility program that lets you edit a disk directory. It is a "freeware" program available on the CUGS disk "disk utilities 1".

This program is actually two programs. The program "DIR. ASSISTANCE" contains instructions and tips for using the program. These can be printed on the screen or a printer. The main program is called "DA V3.0.WOW". This can be loaded directly or from the instruction program.

All activities are performed through menus. The choices are based on loading the directory into memory, editing it and saving the changed directory back to disk. The screen display, with a directory loaded, consists of the first 24 directory entries displayed on the left 2/3 of the screen and the menu on the right. Function keys F1 and F3 allow you to change back and forth between the two menus. A particular editing function is selected by pressing the designated key.

The options available are:

Load a directory - must be done before performing any editing functions.

Save - saves directory back to disk. Changes made are in memory only until they are saved.

Undo - cancel the last change.

Move - move an entry from one place on the directory to another (allows rearranging your listings to put related files together).

sWap - interchange two files in the directory.

Forward - see the next 24 file entries

Backward - see previous 24 file entries

Compact - remove all deleted files in the directory.

Insert - place 0 block spacers between files (5 different lines are given or enter your own characters as spacers).

Replace - change the name of a file, program type or number of blocks.

Delete - remove a directory entry (scratch a file).

Header - change the disk name or id (Note: new id is not written to each track and sector).

Alphabet - alphabetize the entire directory or portion of a directory.

edit - change an entry (allows adding extensions to file names).

bYte - displays file names character by character with a "?" in place of any characters that might disrupt the display (clear screen, color etc.).

Exit - return to BASIC.

drive - permits changing the device and drive number.

~~extensions~~ - add extensions like ,8,1: or ,8: to file names to permit easier loading of files.

uNexten. - remove all extensions to file names.

@disk cmds - wedge disk commands.

The program is easy to use and permits arranging a disk directory into any order. It should be realized that you are CHANGING your disk directory so if you do anything wrong or have a power failure during a SAVE you could lose the programs on the disk. To be safe, use the program on a backup of your disk or save the directory to a new disk before making changes. One other caution - you can insert the same spacer throughout your disk BUT when you want to copy the disk most file copiers cannot handle this and either quit with an error or copy the first spacer only.

The instructions for the program are excellent. The program itself is also very friendly and easy to use. Both the program and instructions are compiled which does increase speed of operation but does not permit making changes to the program to suit your own specific needs. Overall, the program is very useful. Everyone should have it in their disk library. Used carefully, it will allow you to organize your disks to make programs access easier.

CUGS DISK LIBRARY

SCRATCH SAVE

by Earl Brown

COMP UTIL. 8 CH

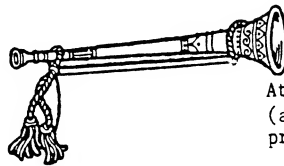
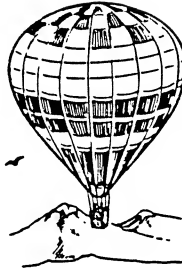
cugs loader
cugs data

func keys.bas
func keys.pal
datapoke aid
load & run.pal
l & r create.c64
time clock
cassette index
helper.boot
helper/64
simple assembler
assembler
editor
loader
translatel.c
ram disk
examiner.boot
basic examiner
multitasker
multitaskr.49152
mis-matcher
function keys
fkeys.49152
info gen
info please
info.49152
iconchanger/geos
keyword construc
data-aid
data-aid.49152
quick key
quick key.49152
mimon (c000)
mimon (7000)
runstop disable
disassembler
str search ldr
string search 2
easy lister
very-able
pres.rs232.c
pres.gizmo.c
memdump828.ins.c
memdump828.c
memdumpdemo.c

CUGS GAZETTE 25

enu
5.<----->
chopper pilot
speedscript 128
ss file conv/128
circle demo
rnd demo
cricket.49152
5.<----->
sketch-pad menu
sketch-pad.ml
printsketch.ml
savesketch.ml

fract'nprac'ce 2
litterbug boot
litterbug.ml
lb1.ml
lb2.ml
gumballrally/128
sprite monitor
sprite mon.20480
sprite mon.49152
datetimestamp128
ss date/time.ml
barricade buster
barricade.ml/64
barricade/128
barricade.ml/128
pizza raid/128
pizza raid.4942
renumber
renumber.49152
crystal boot
crystal clear
crystalcleardemo
top secret
posterprinter128
poster printer
subprograms
subprograms demo
sub1
sub2
5.<----->
geotrashrestorer
word find
power pause
power pause/128
speedscript80/64
sample sketcher
grafnastic.49152
printer changer
printscreen
printscreen.cmp
root race
root race.bas
dynamic screen
dyn scr reader
rapid transit
rapidtransit.cmp
animal match
animal.49152
snake pit
snake pit.49152
crossroads
crossroads.cmp
6.<->
delta war
delta war.obj
sound manager
sound mgr.49152
cats 'n' dogs
cats 'n' dogs.ml
wordcount boot
wordcount gen
wordcount
ss justified
condensed font
cf costumizer
cf costumizr/128
cf costumizr/+16
three pack
contents



Another year's gone and another's started. This one I feel is going to be different. A much better one for all of us. For some reason, I spaced the true meaning of this holiday season too far away from my daily life. Because of this, I plan to be more conscious of my surroundings and communications than I've ever been in the past. I must make a better effort in listening and communicating. What simple device can I employ to download my memory at the proper time without looping as has been the case on too many occasions in the past? I need new information (disks?) new ideas (prgs) and a new direction (not the wrong direction). I want to see this world pull itself together and I want to be a part of it. So much for that.

For the members that buy the GAZETTE magazine, DISK #25 is now available. I have copies still available of #23 and #24. These three disks cover all programs in the issues starting May 1987 through to January 1988. If you neglected to purchase any of the magazines please just scratch the programs you are not entitled to keep.

Also included in our thenew disk listings this month is #CH COMPUTER UTILITIES 8. Perhaps not one of our best utility disks, it does contain a couple of assemblers, a disassembler, and MIMON monitor at two locations (\$C000 and \$7000).

Prize Draw Info:

At each CUGS meeting during 1988 there will be a draw (actually a computer generated draw) for a winner of a prize.

RULES:

Paid up members for 1988 only will be eligible.

Draw will be made at the end of each meeting.

The winner must be present at the meeting to claim the prize. If not, more draws will be made until a winner is found.

All prizes must be accepted as is - no substitutions permitted.

The membership list will be updated at break during each meeting so that new members will be included in the draw.

Prize for January draw - 10 blank diskettes.

MEMBERSHIP NOTE:

Memberships for 1988 are now available.

Cost: \$10.00

Benefits: -access to our disk library

-funds used to:

*update disk library

*purchase hardware for the club

*carry out special activities

-7% discount at Software Supermarket
(note: only new 1988 membership cards will be accepted)

-eligible for 'special prize' draw.

See the treasurer, Harry Chong, or any other executive member.

```

0 POKES3281,0:POKES3280,0:POKE808,234
1 PRINT "*****PLEASE WAIT...LOADING DATA"
2 IFV=OTHENV=1:LOAD"FAFARE.C",8,1
3 POKES6,60:POKES2,60:CLR
4 DIMM$(100),MP$(100)
5 GOSUB10:GOSUB2000:GOSUB3100:
6 PQI="*****"
7 GOTO10000

```

```

10 REM CENTERFOLD
20 REM
30 D=49152
40 FOR A=0TO11:READD
50 IFA=0THENF=PEEK(64)*256+PEEK(63)
70 POK=A,B;B=E+E*B;C=C+1;NEXTA:READD;IFB=ETHEN90
90 E=0:D=D+12:JFC(312)THEN40
110 SYS49152:RETURN
120 DATA169,18,162,192,160,0,120,141,143,2,142,144,1393
130 DATA2,140,49,193,88,96,173,49,193,208,28,165,1384
140 DATA207,208,24,173,141,2,174,50,193,141,50,193,1556
150 DATA236,50,193,240,10,201,5,208,9,141,49,193,1535
160 DATA32,67,192,76,72,235,201,6,208,249,141,49,1528
170 DATA193,32,94,192,76,72,235,173,32,208,141,51,1499
180 DATA193,24,105,8,141,32,208,32,242,192,173,51,1401
190 DATA193,141,32,208,169,0,141,49,193,96,169,0,1391
200 DATA166,214,164,211,141,52,193,142,53,193,140,54,1723
210 DATA193,32,68,229,32,11,193,140,60,193,173,52,1376
220 DATA193,141,55,193,172,55,193,177,107,141,56,193,1676
230 DATA177,111,141,58,193,169,39,56,237,55,193,168,1597
240 DATA177,107,141,57,193,177,111,141,59,193,169,19,1544
250 DATA56,237,52,193,24,109,55,193,168,173,56,193,1509
260 DATA145,105,173,58,193,145,109,169,20,24,109,52,1302
270 DATA193,56,237,55,193,168,173,57,193,145,105,173,1748
280 DATA59,193,145,109,206,55,193,16,179,165,105,24,1449
290 DATA105,40,32,31,193,144,3,32,40,193,206,60,1079
300 DATA193,208,155,238,52,193,173,52,193,201,20,208,1886
310 DATA193,174,53,193,172,54,193,24,32,240,255,76,1605
320 DATA88,192,32,11,193,168,177,105,145,107,177,109,1504
330 DATA145,111,200,208,245,32,40,193,165,106,201,8,1654
340 DATA144,236,96,163,4,162,194,133,106,134,108,169,1655
350 DATA216,162,198,160,25,133,110,134,112,163,0,133,1552
360 DATA105,133,107,133,109,103,111,96,230,106,230,108,1601
370 DATA230,110,230,112,96,0,0,0,0,0,0,0,0,0,0,778

```

```

→ 1500 REM *** DROP IN ***
1510 W1=1:L=LEN(ME$)
1520 PRINT "MI"
1530 FORN=1 TO W1:CH$=MID$(ME$,W1,1):PRINTTAB(5)"MI" *CH$:GOSUB1570:NEXT:
1540 GOSUB1570:PRINT "MI" *MI":GOSUB1570:PRINTTAB(2)CH$
1550 IF W1<LEN(ME$) THEN W1=W1+1:PRINT:PRINT".I.I";:GOTO1530
1560 PRINT "MI":RETURN
1570 FORM=1 TO S:NEXT:RETURN

```

```

1905 REM SPECIAL COLORED BORDER EFFECT
2001 S=52800:REM *** CHANGE S TO RELOCATE
2002 PRINTCHR$(147):PRINT:FORJ=S/STOS*20
2003 READA:POKEJ,A:X=X+A:NEITJ
2004 RETURN
2005 DATA32,228,255,240,3,133,251,96,238,32,208,162
2006 DATA90,202,208,253,141,18,208,240,235

```

```

2009 REM CURTAIN EFFECT TO CLEAR SCREEN
3100 LOC = 52900: REM RELOCATABLE
3125 FORT=0T0103:READA:CH=CHA:POKELOC+T,A:NEXT
3135 REM
3140 RETURN
3200 DATA169,0,72,169,0,133,251
3201 DATA169,4,133,252,169,0,133
3202 DATA253,169,216,133,254,162,0
3203 DATA160,1,177,253,72,177,253
3204 DATA136,145,251,104,145,253,200
3205 DATA200,192,20,208,239,136,169
3206 DATA32,145,251,160,38,177,253
3207 DATA72,177,251,200,145,251,104
3208 DATA145,253,136,136,192,18,208
3209 DATA239,24,165,251,105,40,133
3210 DATA251,165,252,105,0,133,252
3211 DATA24,165,253,105,40,133,253
3212 DATA165,254,105,0,133,254,232
3213 DATA224,25,208,182,104,168,200
3214 DATA152,192,20,208,155,96

```

```

10002 REM TITLE SCREEN -3 SEC.
10003 REM
10004 CS$="POKE53280,0:POKE53281,0
10005 PRINT" WELCOME TO OUR
10010 FORI=1TO10:PRINTHD$(CS$,I,1)+" WINNING WAYS"
10012 NEXT
10015 FORI=1TO1000:NEXT:SYS52900
10016 REM
10017 REM
10018 REM MENU SELECTION: KEY SCREEN
10019 REM
10020 PRINT"
10030 PRINT"
10040 PRINT"
10050 PRINT"

```

```

10054 REM THREE CHOICES - ADD NEW NAMES
10055 REM MARK ATTENDANCE/PICK A WINNER
10060 GETK$:IFK$=""THEN10060
10070 IFK$<>"A"ANDK$<"M"ANDK$<"P"THEN10060
10080 IFK$="M"THENPRINTP$LOADING DATA:GOSUB13000:PRINT""
10090 IFK$="A"THENGOSUB12000:PRINT""
10090 IFK$="P"THENGOSUB14000:PRINT""

```

```

10094 REM GET AND DISPLAY A WINNER!
10101 IF#0(1)=="THENPRINT FIRST ... CHECK WHO'S HERE!":GOSUB13000
10105 IF#0(1)=="ANDNP=0THEN10000
10110 PRINT"
```

```

10130 WN=INT(WN*P)+1
10133 REM 816 SHOW WITH MUSIC!
10140 ME$="THE WINNER IS... "+MP$(WN)
10150 SYS23965+155
10155 GOSUB1000:PRINTPO$(MP$(WN))
10156 SYS2800:POKE198,Q1GETK$:IFK$="X"THEN10158
10157 GOTO10156
10158 SYS52900:POKE23965+1079,5;

```

```

10159 REM ***** THE DESTROYER! *****
10160 SYS64738

12002 REM ADD NEW NAMES
12005 GOSUB50000
12007 OPEN3,B,3,"MENS,S,A"
12010 PRINT:PRINTENTER NEW NAMES: (TYPE 'DONE' TO END)
12020 PRINT:PRINT(FIRST NAME LAST NAME)
12030 INPUT"ENTER NEW NAME: ###";NN$
12040 PRINT:PRINT IS THIS CORRECT?(Y/N);
12050 GETK$;IFK$="Y"THEN12050
12051 IFNN$="DONE"THEN12140
12055 IFK$="Y"ORX$=CHR$(13)THEN12100
12060 PRINT:PRINT:PRINTFORI=1TO3:PRINT " ";NEXT:PRINT
12067 PRINT:PRINT

```

```
12100 IFNN$="*THEN12140
```

```

12101 PRINT#3,M#S:PRINT
12110 PRINT"MORE (Y/N)?";
12120 GET#1:IF#1=""THEN12120
12130 IF#1="Y"OR#1=CHR$(13)THEN12010
12140 CLOSE3:CLOSE15:RETURN
13003 REM TAKE ATTENDANCE!
13008 GOSUB50000
13010 OPEN3,8,3,"MEMS,S,R"
13020 R=1
13030 GET#3,G#1:IF#1=""THEN13030
13035 IF#1=CHR$(13)THEN 13100
13040 IFST<>0)THENCLOSE3:CLOSE15:GOTO13200
13050 M#S=M#S+G#1:GOTO13030
13100 M$(R)=M#S:M#S="":R=R+1
13110 GOTO13030
13200 PRINT"MARK THOSE PRESENT: ";S=21
13210 IF R <=20 THEN S=1:GOTO13300
13220 FOR=1 TO20
13225 A$=RIGHT$(STR$(A),LEN(STR$(A))-1):IFLEN(A$)<2)THEN A$=" "+A$
13230 M#S=A$:PRINTM#S;"LEFT$(M$(A),16)
13240 NEXT
13300 PRINT"END"
13320 FOR A=S TO R-1
13325 A$=RIGHT$(STR$(A),LEN(STR$(A))-1):IFLEN(A$)<2)THEN A$=" "+A$
13330 M#S=A$:PRINTTAB$(S-1);M#S;"LEFT$(M$(A),16)

```

```

13340 WEIT
13390 REM
13392 REM
13394 REM ATTENDANCE LOOP
13395 REM
13396 REM
13398 REM
14000 D=1
14001 PRINT"XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXSELECT NAMES BY NUMBER(0=DONE)";P$=""
14010 INPUT$
14020 IF P$="" THEN I=14100
14030 REM
14100 IF LEN(P$)>2 THEN PRINT:PRINT"NO SUCH NUMBER!";FOR PA=1 TO 400:WEIT:GOTO 13200
14102 IF VAL(P$)R-1 THEN PRINT:PRINT"TOO BIG!";FOR A=1 TO 400:WEIT:GOTO 13200
14110 P=VAL(P$);IF P=0 THEN B=B+1:GOTO 15000
14120 NP$(D)=M$(P);P$="":D=D+1
14125 PRINT"XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX" (36 Spaces)" ";
14130 GOTO 14001
15000 Z=1:A$=""
15001 FOR Y=1 TO D:IF MP$(Z)=MP$(Y) AND Z<>Y THEN TAG=Y
15002 NEXT Y:Z=Z+1:IF Z<D THEN 15001
15003 IF TAG=1 THEN PRINT"XXXXXXXXXXXXXXXXXXXXDuplicate name entered!! PLEASE REDO!";GOTO 13000
15005 NP=D:D=1:W=RND(-TI):RETURN
50000 REM
50001 REM OPEN AND CHECK ERROR CHANNEL
50003 REM
50005 REM
50006 OPEN #5,8,15:RETURN
51000 INPUT#1S,A$,B$,C$,D$
51100 IF A$(">") THEN CLOSE #3:CLOSE #5:PRINT A$,B$:END
52000 CLOSE #5:RETURN

```


This is the third article in a series examining different file types and the programming involved with each. The first article dealt with the different file types and how a disk is set up to handle them. The second article examined PROGram files. This article examines SEquential files.

SEquential files are primarily used for saving data. The data is stored to disk from within a BASIC program and accessed within a program. The main uses for SEquential files are:

- 1) in large programs, memory is conserved by having the data in a file which is accessed and put into variables when needed. This prevents a "duplication" of storage space caused by having data in DATA statements AND in variables.
- 2) when lots of variable data is going to be used, it can be stored in SEquential files and accessed as needed. This eliminates the need to have ALL data stored in the computer at one time.
- 3) in programs where listing the program and seeing the data (e.g. words for a HANGMAN game) could ruin the effect of the program, the data can be kept in a SEquential file and be generally unavailable to prying eyes.

Before examining the programming involved with SEquential files, I'll examine the advantages and disadvantages of using this file type compared to other methods for storing data. One advantage is that programming SEquential files is easier than programming RELative files or random access files. A second advantage is that SEquential files store data more compactly on disk than RELative files. A major DISadvantage is that data must be accessed from the start each time. For a large file this could involve a fair amount of time to get to a certain data item near the end of the file. Another disadvantage is that the data is separate from the program and cannot as easily be transferred to another disk or changed as could data statements.

Programming with SEquential files involves 3 steps - OPENing a line of communication to the disk drive, transferring data, and CLOSEing (turning off) the line.

OPENing a line of communication involves a specific statement. The general format is: OPEN 1,8,5,"O:FILENAME,S,W". The BASIC keyword "OPEN" is used to tell the disk drive that you want to communicate with it. The three numbers ("1,8,5") have the following significance: the first number (LOGICAL file number) can be any value between 1 and 255. It is used for all reference to the file. It is only important when more than one sequential file is open at one time. It is the way the computer keeps track of which file it is accessing. If more than one file is open at once, each must have its own unique LOGICAL file number.

The second number (DEVICE number) is the device number for the disk drive. For a file on a second disk drive substitute 9 for the 8. Use a "1" for "8" for cassette tape.

The last number (SECONDARY address) can be any value from 0 to 15, although 0 and 1 are specifically for READ and WRITE operations and 15 is the disk COMMAND channel and generally avoided unless needed. This leaves values between 2 and 14. It doesn't matter what you select for a secondary address except different values for each file must be used if two or more files are open at the same time.

Within the quotes are found "O:FILENAME,S,W". The "O:" is the drive number. It is used if you

have a dual disk drive (like a 4040 or 8050) but is thought to help eliminate the infamous "save/replace" bug. It can be omitted for a single drive. The FILENAME is the name you use for the stored data. This name follows the rules for all file names and must be unique on a disk. Following the filename is a comma and a letter indicating the type of file ("S" is file type SEquential). Other letters that are used are "U" (USEr), P (PROGram). Later I will illustrate the differences involved in using these letters. Following this is another comma and a letter. This letter may be W (WRITE), R (READ), or A (APPEND). The file type and access letter may be omitted for READING a file from the disk but MUST be included when WRITING or APPENDING to a disk. Some examples of the use of the OPEN statement:

OPEN 2,8,2,"CUGS,S,W" = OPEN a line (2) of communication to the disk drive to (W)rite a (S)equential file called "CUGS"

OPEN 3,8,6,"CUGS DATA" = OPEN a file called "CUGS DATA" to READ the data it contains (equivalent to OPEN 3,8,6,"CUGS DATA,S,R").

OPEN 8,8,8,"CUGS DATA,S,A" = OPEN the file called "CUGS DATA" to (A)ppend more data to the END of the current data.

Note: if you are going to have two or more files open at the same time, it is easier to keep track of them and prevent mistakes if you use the same numbers for the logical file number and secondary address for each file. i.e. OPEN 2,8,2,"CUGS 1":OPEN 3,8,3,"CUGS 2" has two files open with "2" being used for the file "CUGS 1" and "3" used for "CUGS 2".

TRANSferring data. Once the line of communication is OPENed, the next step is to transfer data. Data may be written to or retrieved from the file. To Write or Append to a file the BASIC keyword PRINT# is used. To GET data from a file, you can use either INPUT# or GET#.

The full format of the PRINT# statement is: PRINT#2,A\$. This stores the current value of A\$ in a file that was OPENed to Write or Append with logical file number 2. The logical file number MUST be the same used in the OPEN statement. Often the data is accessed and written to the disk in a loop containing the PRINT# statement. This results in the data being stored on disk SEquentially with a carriage return [chr\$(13)] written after each data item. The data becomes like cars of a train with a RETURN separating one item from the next.

To get data from a file you use the INPUT# or GET# statement. Both are used with files that have been OPENed to Read data. The difference between them is that INPUT# will take all the data up to a carriage return as ONE data item while GET# will get one character at a time. The way the data will be used in the program determines which will be used. These statements have the following format: INPUT#2,A\$ or GET#2,A\$. Both are accessing the file OPENed using logical file number 2. In the first case A\$ could contain "RICHARD MAZE" while in the second case A\$ would contain only "R".

CLOSEing the line of communication turns off the link to the disk drive. A file should always be closed after being accessed. The format is: CLOSE 2 - the BASIC word CLOSE followed by the logical file number used in the OPEN statement.

Next month I'll examine different ways to store and accessing sequential data, error checking and accessing files of undetermined length. This will include some tricks you can use to make data (and your programs) inaccessible to

S
I
R

R
I
C

H
A
R
D
S

B
A
S
I
C

SOFTWARE REVIEW:

REVIEW: DISK MAINTENANCE

DISK MAINTENANCE, as the name suggests, permits performing a number of operations on a disk directory. It is a "freeware" program available on the CUGS disk "DISK UTILITIES 2".

The program will automatically run. After loading "dm-boot" (with ",8,1") the program displays an introductory screen, a three-choice menu - 1) DISK MAINTENANCE 2) INSTRUCTIONS, and 3) A COMMERCIAL MESSAGE.

The instructions are well done, with a good, clear description of each command available. The commercial message describes an enhanced version of DISK MAINTENANCE which can be obtained from the programmers (for a price). According to the commercial, the enhanced version contains a "state-of-the-art" sector editor as well as a program compactor/uncompactor. It also permits editing the BAM and contains additional maintenance commands.

When DISK MAINTENANCE is selected from the main menu, a submenu - 1) DISK MAINTENANCE FW and 2) RESET. RESET returns you to BASIC. When you press 1, you will get the prompt: INSERT DISK THEN PRESS ANY KEY. The program will read the directory of the disk and displays it on the screen. The cursor keys can now be used to scroll through the directory. The commands available are:

- A - ADDRESSES...display the start and end memory addresses of the selected file.
- B - VIEW BAM...graphically displays the block availability map.
- C - CHANGE DISKS...used to view and edit another disk.
- E - EXCHANGE FILES...permits swapping directory locations of two files. This allows the grouping of related files together on a disk.
- L - LIST FILE AS BASIC PROGRAM...the selected file is listed on the screen. This gets around special characters (like shifted L's or deletes after a REM) that could prevent or change a normal listing.
- N - NAME DISK...permits changing a disk name. This will allow the inclusion of special control characters in the name if you want.
- P - APPEND TWO BASIC FILES...permits creating one file on disk by hooking two other files together. The second file is added to the end of the first.
- R - RENAME FILE...permits changing the name of a file.
- S - SCRATCH FILE...allows removing a file from the disk.
- T - TEST TWO BASIC PROGRAMS...compares selected programs and displays differences between them.
- V - VALIDATE DISK...performs the standard validate command.
- Z - SEQUENTIAL FILE LISTER...displays any file as a sequential file on the screen.

Besides the above commands, DISK MAINTENANCE also permits unscratching deleted files, changing a file type, and locking/unlocking a file. DISK MAINTENANCE lists all files in a directory including deleted files. You can recover deleted files by pressing CURSOR RIGHT and then CURSOR UP or CURSOR DOWN to change the file type from DEL to the type you want. This same method is used to change the type of file on the disk. After pressing cursor right, the < key is used as a toggle to lock/unlock files. A locked file cannot be scratched by the normal scratch command. A '*' file is an improperly closed file, removed by using the validate command. To view a '*' file, press "*" after the cursor right and

the pointer will be corrected. You can now view the file up to the point of the error.

DISK MAINTENANCE permits performing a number of operations that are not normally available or very difficult to do. This one program can be very useful particularly when you want to examine files on a disk or change the directory entries of files.

COMPUTERFEST

(Sponsored by the Apple Users' Group)
Vagabond Motor Inn 11 am - 5 pm

FEBRUARY 7, 1988

Displays by major retailers of hardware
and software!

User groups will each have a display area!

Flea market of computer items!

NO ADMISSION CHARGE!

If you wish to sell something at the "flea market" (hardware, original software, books, etc.), goods must be registered by noon. The Apple group takes 10% of the sale price or \$1.00 (whichever is more). Money or unsold items are to be picked up from 5-6 pm.

